WHAT IS CLAIMED IS:

1. A method for managing railcar movement in a railyard based on the flow of railyard tasks, using a system including a computer that includes a processor, a memory device, and a database, the railyard including six subyards, said method comprising the steps of:

inputting initial parameters to the computer;

simulating railyard task flow utilizing a yard performance model and the initial parameters; and

determining if a train schedule can be met based on the simulated yard task flow.

2. A method in accordance with Claim 1 wherein the six subyards include a surge yard, a receiving yard, a receiving inspected (RI) yard, a classification yard, a departure yard and a departure inspected (DI) yard, said step of inputting initial parameters to the computer comprises the steps of:

inputting a simulation ending time (T_f);

inputting at least one train schedule parameter;

inputting a time dependent modulation (TDM) parameter;

inputting at least one initial task backlog parameter;

inputting at least one yard topology parameter; and

inputting at least one initial labor assignment parameter.

3. A method in accordance with Claim 2 wherein the railyard tasks include at least one of surge-to-receive railcar movement, receive yard inspection, receive-to-classification railcar movement, classification-to-departure railcar movement, and departure yard inspection, said step of simulating railyard task flow utilizing a yard performance model includes the step of executing a railyard performance algorithm.

4. A method in accordance with Claim 3 wherein said step of executing a railyard performance algorithm comprises the steps of:

initializing a time (T) equal to the clock time at the railyard in which the simulation is to begin;

updating task backlogs based on the initial parameters;

computing task flow rates based on the updated task backlogs; and

determining whether time (T) equals simulation ending time (T_f).

5. A method in accordance with Claim 4 wherein said step of determining if a train schedule can be met comprises the steps of:

verifying that time (T) does equal simulation ending time (T_f); and

calculating a train schedule based on the computed task flow rates.

6. A method in accordance with Claim 4 wherein said step of simulating railyard task flow further comprises the steps of:

verifying that time (T) does not equal simulation ending time (T_f); and

executing a railyard performance algorithm subroutine.

7. A method in accordance with Claim 6 wherein said step of executing a subroutine further comprises the steps of:

re-updating task backlogs;

re-computing task flow rates based on the re-updated task backlogs; and

determining whether time (T) equals simulation ending time (T_f).

8. A method in accordance with Claim 7 wherein said step of re-updating task backlogs comprises the steps of:

determining whether time (T) equals a full hour;

determining whether time dependent modulation has been selected;

determining whether any of the subyards are congested; and

incrementing time (T) by 15 minutes.

9. A method in accordance with Claim 8 wherein said step of determining whether time (T) equals a full hour comprises the steps of:

determining a new labor mix;

incrementing time (T) by fifteen minutes; and

activating the new labor mix.

10. A method in accordance with Claim 8 wherein said step of determining whether time dependent modulation (TDM) has been selected comprises the step of modifying all task rates.

11. A method in accordance with Claim 8 wherein said step of determining whether any of the subyards are congested comprises the step of modifying task rates of tasks performed by an engine crew.

12. A method in accordance with Claim 7 wherein said step of determining whether time (T) equals simulation ending time (T_f) comprises the steps of:

verifying that time (T) does not equal simulation ending time (T_f); and

repeating execution of the subroutine.

13. A method in accordance with Claim 7 wherein said step of determining whether time (T) equals simulation ending time (T_f) comprises the steps of:

verifying that time (T) does equal simulation ending time (T_f);

discontinuing execution of the subroutine; and

calculating a train schedule based on the re-computed task flow rates.

14. A system for managing railcar movement in a railyard based on the flow of railyard tasks, said system comprising a computer comprising a processor, a memory device, and a database, the railyard including at least one of a surge yard, a

receiving yard, a receiving inspected (RI) yard, a classification yard, a departure yard and a departure inspected (DI) yard, said system configured to:

input initial parameters to said computer,

simulate railyard task flow utilizing a yard performance model and the initial parameters; and

determine if a train schedule can be met based on the simulated yard task flow.

15. A system in accordance with Claim 14 wherein to input the initial parameters to said computer, said system further configured to:

input a simulation ending time (T_f);

input at least one train schedule parameter;

input a time dependent modulation (TDM) parameter;

input at least one initial task backlog parameter;

input at least one yard topology parameter; and

input at least one initial labor assignment parameter.

16. A system in accordance with Claim 15 wherein to simulate railyard task flow utilizing the yard performance model, said system further configured to execute a railyard performance algorithm.

17. A system in accordance with Claim 16 wherein to execute said railyard performance algorithm, said system further configured to:

initialize a time (T) equal to the clock time at the railyard in which the simulation is to begin;

update task backlogs based on the initial parameters;

compute task flow rates based on the updated task backlogs; and

determine whether the time (T) equals the simulation ending time (T_f).

18. A system in accordance with Claim 17 wherein to determine if a train schedule can be met, said system further configured to:

verify that the time (T) equals the simulation ending time (T_f); and

calculate a train schedule based on the computed task flow rates.

19. A system in accordance with Claim 17 wherein to simulate railyard task flow, said system further configured to:

verify that the time (T) does not equal the simulation ending time (T_f); and

execute a railyard performance algorithm subroutine.

20. A system in accordance with Claim 19 wherein to execute said subroutine, said system further configured to:

re-update the task backlogs;

re-compute the task flow rates based on the re-updated task backlogs; and

determine whether the time (T) equals the simulation ending time (T_f).

21. A system in accordance with Claim 20 wherein to re-update task backlogs, said system further configured to:

determine whether the time (T) equals a selected length of time;

determine whether time dependent modulation has been selected;

determine whether any of the subyards are congested; and

increment the time (T) by a fraction of the selected length of time.

22. A system in accordance with Claim 21 wherein to determine whether said time (T) equals the selected length of time, said system further configured to:

determine a new labor mix;

increment the time (T) by the fraction of the selected length of time; and

activate said new labor mix.

23. A system in accordance with Claim 21 wherein to determine whether time dependent modulation (TDM) has been selected, said system further configured to modify all task rates.

24. A system in accordance with Claim 21 wherein to determine whether any of the subyards are congested, said system further configured to modify task rates of tasks performed by an engine crew.

25. A system in accordance with Claim 20 wherein to determine whether the time (T) equals the simulation ending time (T_f), said system further configured to:

verify that the time (T) does not equal the simulation ending time (T_f); and

repeat execution of said subroutine.

26. A system in accordance with Claim 20 wherein to determine whether the time (T) equals the simulation ending time (T_f), said system further configured to:

verify that the time (T) equals the simulation ending time (T_f);

discontinue execution of said subroutine; and

calculate a train schedule based on the re-computed task flow rates.

27. A railyard performance model for use in management of a railyard, the railyard including six subyards including a surge yard, a receiving yard, a receiving inspected (RI) yard, a classification yard, a departure yard and a departure inspected (DI) yard, said model configured to:

simulate railcar movement in a railyard based on the flow of railyard tasks; and

determine if a train schedule can be met based on the simulation.

28. A model in accordance with Claim 27, wherein to simulate railcar movement, said model further configured to:

initialize a time (T) equal to the clock time at the railyard in which the simulation is to begin;

update task backlogs based on a set of initial parameters;

compute task flow rates based on the updated task backlogs; and

determine whether the time (T) equals a simulation ending time (T_f).

29. A model in accordance with Claim 28 wherein to determine if a train schedule can be met, said model further configured to:

verify that the time (T) equals the simulation ending time (T_f); and

calculate a train schedule based on the computed task flow rates.

30. A model in accordance with Claim 28 wherein to simulate railyard task flow, said model further configured to:

verify that the time (T) does not equal the simulation ending time (T_f); and

execute a railyard performance algorithm subroutine.

31. A model in accordance with Claim 30 wherein to execute said subroutine, said model further configured to:

re-update the task backlogs;

re-compute the task flow rates based on the re-updated task backlogs; and

determine whether the time (T) equals the simulation ending time (T_f).

32. A model in accordance with Claim 31 wherein to re-update task backlogs, said model further configured to:

determine whether the time (T) equals a selected length of time;

determine whether time dependent modulation has been selected;

determine whether any of the subyards are congested; and

increment the time (T) by a fraction of the selected length of time.

33. A model in accordance with Claim 32 wherein to determine whether the time (T) equals the selected length of time, said model further configured to:

determine a new labor mix;

increment the time (T) by the fraction of the selected length of time; and

activate the new labor mix.

34. A model in accordance with Claim 32 wherein to determine whether time dependent modulation (TDM) has been selected, said model further configured to modify all task rates.

35. A model in accordance with Claim 32 wherein to determine whether any of the subyards are congested, said model further configured to modify task rates of tasks performed by an engine crew.

36. A model in accordance with Claim 31 wherein to determine whether the time (T) equals the simulation ending time (T_f), said model further configured to:

verify that the time (T) does not equal the simulation ending time (T_f); and

repeat execution of the subroutine.

37. A model in accordance with Claim 31 wherein to determine whether the time (T) equals the simulation ending time (T_f), said model further configured to:

verify that the time (T) equals the simulation ending time (T_f);

discontinue execution of the subroutine; and

calculate a train schedule based on the re-computed task flow rates.